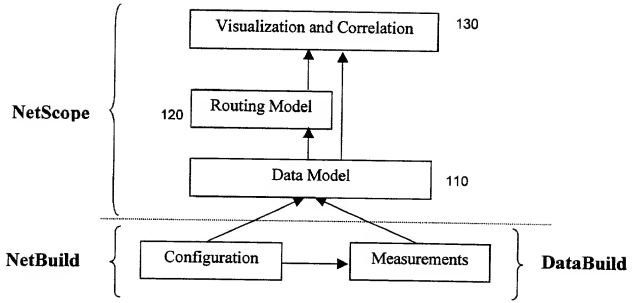
Visualization and Correlation
130
NetScope
120
Routing Model
Data Model
110
NetBuild
Configuration
Measurements
DataBuild

Figure 1

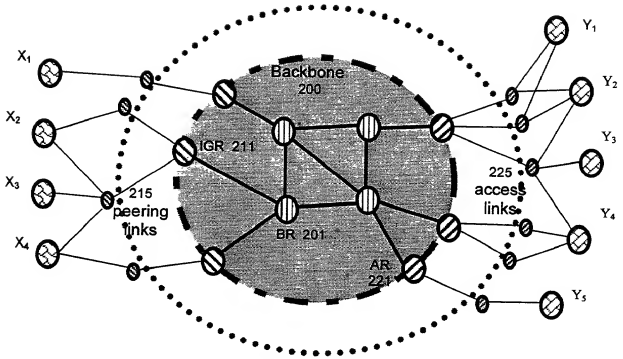
X1
X2
X3
X4
Backbone
200
IGR 211
215
peering
links
BR 201
AR
221
225
access
links
Y1
Y2
Y3
Y4
Y5

Figure 2

```
For each flow: (input, dest, start, finish, bytes)
        dest_prefix = longest_prefix_match(dest, dest_prefix_set);
        egress_set = reachability(dest_prefix);
        start_bin = [start/width] * width;
        finish_bin = [finish/width] * width;
        if (start_bin == finish_bin)
                volume[input, egress-set, start_bin] += bytes;
        else   /* Compute volume of traffic for each time_bin */
                byte_rate = byutes / (finish – start)
                volume[input, egress_set, start_bin] += byte_rate * (start_bin + width – start);
                for (time_bin = start_bin + width; time_bin < finish_bin; time_bin += width)
                        volume[input, egress_set, time_bin] += byte_rate * width;
                volume[input, egress_set, finish_bin] += byte_rate * (finish – finish_bin);
Output for each aggregate: (input, egress_set, time_bin, volume)
```

Figure 3

```
For each flow: (input, output, src, dest, start, finish, bytes)
        dest_prefix = longest_prefix_match(dest, dest_prefix_set);
        egress_set = reachability(dest_prefix);
        if (input.type == peer)   /* Inbound or (ingress) transit flow */
                compute volume[input, egress-set, input, output, time_bin] for each bin;
        else   /* Outbound or (egress) transit flow */
                src_prefix = longest_prefix_match(src, src_access_prefix_set);
                if (src has no match)
                  ingress_set = sendability(src_prefix);
                  compute volume[ingress_set, egress_set, input, output, time_bin] for each bin;
Output for each aggregate: (ingress_set, egress_set, input, output, time_bin, volume)
```

Figure 4

```
For each aggregate:  (ingress_set, egress_set, input, output, time_bin, volume)
        For each a in ingress_set
                route = Route(a, egress_set);
                if (route does not use input and output links)
                        remove a from ingress_set;
        if (ingress_set ≠ ∅)
                for each a in ingress_set
                        dvolume[a, egress_set, time_bin] += volume / size_of(ingress_set);
        else
                count as a miss;
Output for each demand:  (a, egress-set, time_bin, dvolume)
```

Figure 5

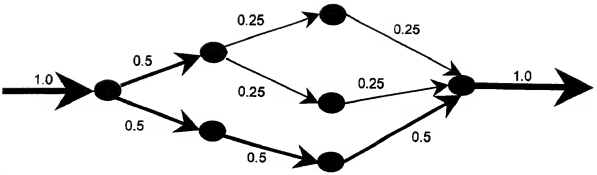

Figure 6

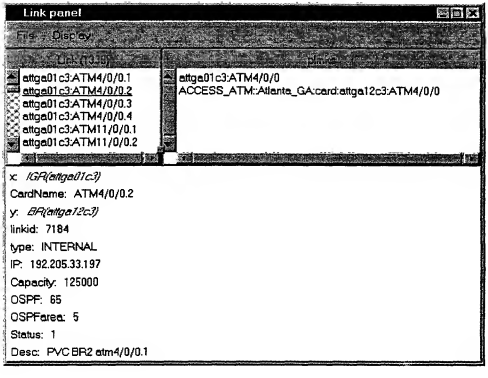
Link panel
File Display
Link (1336)
attga01c3:ATM4/0/0.1
attga01c3:ATM4/0/0.2
attga01c3:ATM4/0/0.3
attga01c3:ATM4/0/0.4
attga01c3:ATM11/0/0.1
attga01c3:ATM11/0/0.2
Name
attga01c3:ATM4/0/0
ACCESS_ATM::Atlanta_GA:card:attga12c3:ATM4/0/0
x: IGR(attga01c3)
CardName: ATM4/0/0.2
y: BR(attga12c3)
linkid: 7184
type: INTERNAL
IP: 192.205.33.197
Capacity: 125000
OSPF: 65
OSPFarea: 5
Status: 1
Desc: PVC BR2 atm4/0/0.1

Figure 7

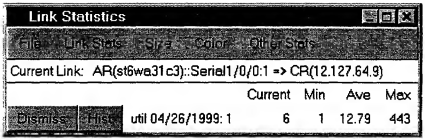

Figure 8

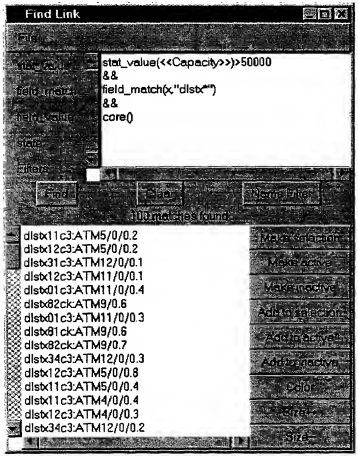

Figure 9

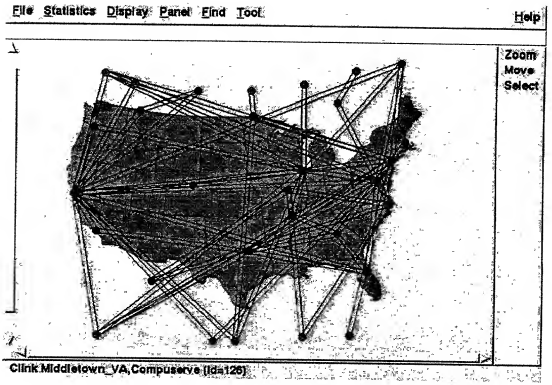
File Statistics Display Panel Find Tool
Help
Zoom
Move
Select
Clink Middletown_VA,Compuserve (Ids126)

Figure 10

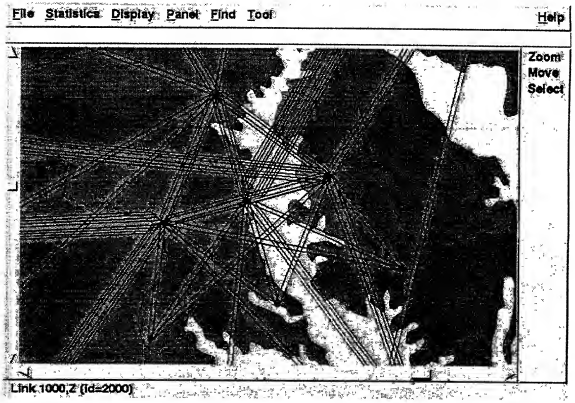
File Statistics Display Panel Find Tool
Help
Zoom
Move
Select
Link 1000,2 (id=2000)

Figure 11